MATHEMATICAL DECOMPOSITION AND SIMULATION IN REAL-TIME PRODUCTION SCHEDULING

Wayne Davis
Department of General Engineering
University of Illinois
Urbana, Illinois 61801 USA

Albert Jones
Center for Manufacturing Engineering
National Bureau of Standards
Gaithersburg, Maryland 20899 USA

ABSTRACT

This paper discusses an on-line, real-time production scheduling algorithm for automated manufacturing systems. Decomposition theory is used to transform a multi-criteria, production scheduling problem from a block angular structure into a two-level hierarchical structure. The top level, called the supremal, considers a list of jobs, due dates, precedence constraints, and objectives. It generates a set of potential scheduling rules and evaluates those rules using an on-line, distributed simulation package. The supremal outputs a list of tasks with proposed start and finish times to each of the lower level systems under its control. Each lower level system, called an infimal, then uses a similar simulation approach to sequence those tasks and generate actual start and finish times. These times, together with status on all other tasks, provide the feedback needed by the supremal to close the control loop.

INTRODUCTION

Recently, manufacturing companies have invested large sums of money in advanced production facilities. These facilities include computer hardware and software, robotics, automated material handling, machining centers, and data management and communications equipment. They can be procured from a single vendor, as an integrated system, or from a series of vendors and integrated by the user.

The primary motivation for these investments has been the perceived notion that productivity, and hence profits, will increase. This increase would result from better quality, faster throughput, larger machine utilization, and fewer production problems. For the most part, these expectations have not been met. One of the primary reasons is the difficulty involved in scheduling and controlling activities in the dynamic environment on the shop floor. And, the increased flexibility provided by these new technologies has magnified, rather than reduced, these problems.

Several hierarchical models (8) have been proposed for controlling shop floor activities. These models decompose manufacturing functions into levels and specify the interfaces between levels. Scheduling is typically assigned to one level and is still considered an off-line function. Graves (6), and Raman (14) are excellent sources on classical off-line scheduling algorithms.

Davis (3), Stecke (15), Jackson and Jones (7), and others have argued the merits of a real-time production scheduler. This paper proposes a two-level, real-time scheduling/control technique, based on the decomposition approach to mathematical programming. A detailed algorithm is provided for developing schedules and methods are proposed for the real-time execution of those schedules.

Specifically, the next section gives a brief overview of the theory of decomposition in mathematical programming. These concepts are applied to the proposed production scheduling problem in Section 3. The proposed algorithm is described in section 4, and future work is outlined in section 5.

THE USE OF A MATHEMATICAL DECOMPOSITION

The Problem

A formal statement of the production scheduling (PS) problem is as follows: Assume that JOB_j $(j=1,...,J)$ have been issued to the PS with associated due dates D_j $(j=1,...,J)$ and that JOB_j requires the production of a specific product $PROD_m$ $(m=1,...,M)$. And, assume that the processes P_n $(n=1,...,N)$ are available, and that $TASK_{kjn}$ $(k=1,...,K)$ are the tasks of JOB_j to be performed on

P_n. Then, if we define

E_{kjn} ($k=1,\ldots K$; $j=1,\ldots,J$; $n=1,\ldots,N$)

as the earliest start time for P_n upon TASK_{kjn} from JOB_j and

L_{kjn} ($k=1,\ldots,K$; $j=1,\ldots,J$; $n=1,\ldots,N$)

as the latest finish time for P_n upon TASK_{kjn} from JOB_j the production scheduling problem is to optimize the utility function

$$W[f^1(E_{111},\ldots,L_{kjn}),\ldots,f^L(\cdot)]$$

(subject to due date, material handling, resource availability, precedence constraints, and alternate routings) where $f^l(\cdot)$ for $l=1,\ldots,L$ are the criteria to be considered in the optimization. They typically include minimizing tardiness, maximizing production throughput, maximizing process utilization, etc.

By reordering the constraints, it is possible to generate a block angular form for the mathematical programming formulation of the PS (see Figure 1a). Let $\mathbf{x}$ represent the vector of decision variables. Partition $\mathbf{x}$ into a set of subvectors $\mathbf{x}_i$ ($i=1,\ldots,N$) where each $\mathbf{x}_i$ will be assumed to contain n_i components. The partitioning generates N+1 blocks of constraints: $X_0 = \{\mathbf{x} | g_0(\mathbf{x}_1,\ldots,\mathbf{x}_N) \leq b_0\}$ and $\mathbf{X}_i = \{\mathbf{x}_i | g_i(\mathbf{x}_i) \leq b_i\}$ ($i=1,\ldots,N$). X_0 represents the set of points satisfying the set of coupling constraints $g_0(\cdot)$. X_i is defined via m_i functional constraints $g_i(\cdot)$ involving the components of $\mathbf{x}_i$ only.

As noted above, the PS problem must be solved in the context of a hierarchical decision-making and control architecture. However, the structure shown in Figure 1a does not readily fit into this framework. To capture the desired hierarchical interaction, we will extend the decomposition approach of single criterion, constrained, mathematical programming problems. We now present a brief review of the fundamental principles of this technique.

The Approach

Mathematical decomposition procedures (5,9) are used to change the block angular decision structure in Figure 1a into the hierarchical decision structure depicted in Figure 1b. As illustrated, that hierarchical structure contains N+1 decision-making elements. At the upper level, the supremal explicitly considers the coupling constraints X_0. The lower level contains N infimals. Each considers the individual constraints within the decision subspace X_i ($i=1,\ldots,N$) defined by $g_i(\mathbf{x}_i)$. The decomposition procedure must also specify the interaction between the two levels of the hierarchy so that the solution to the overall problem, given as

$$\begin{aligned}&\min \ f(x_1,\ldots,x_N) && \text{PO}\\ &\quad \text{s.t. } (x_1,\ldots,x_N) \in \{X_0 \cap X_1 \cap \ldots \cap X_N\}\end{aligned}$$

can be achieved.

To solve PO, the supremal builds an approximation PO', which is given as

$$\begin{aligned}&\min \ f(x_1,\ldots,x_N) && \text{PO'}\\ &\quad \text{s.t. } (x_1,\ldots,x_N) \in \{X_0 \cap X'_1(t) \cap \ldots \cap X'_N(t)\}\end{aligned}$$

For each infimal i, PO' includes a reference model $X'_i(t)$ on iteration t of the decision subspace X_i. The manner in which these reference models are constructed is dependent upon the decomposition approach being adopted and is discussed later in the presentation.

Let $x^*_i(t)$ (i=1,...,N) represent the optimal solution to the PO' on iteration t. Using this solution, the supremal must develop the information necessary to direct the infimals' decision-making on the next iteration t+1. Let $\gamma_i(t+1)$ represent the coordinating information to be presented to the infimal i on iteration t+1. Further assume that there exists a set of functions $\Gamma_i(\cdot)$ for i=1,...,N such that

$$\gamma_i(t+1) = \Gamma_i[x^*_1(t),\ldots,x^*_N(t)] \quad (i=1,\ldots,N) \tag{1}$$

Given this $\gamma_i(t+1)$, the infimal i then formulates its decision as

$$\begin{aligned}&\min \ f_i[x_i(t+1) \mid \gamma_i(t+1)] && \text{Pi}\\ &\quad \text{s.t. } g_i[x_i(t+1) \mid \gamma_i(t+1)]\end{aligned}$$

where $f_i[x_i(t+1)]$ is the component of the overall objective function $f[x_1,\ldots,x_N]$ dealing with the elements of x_i only.

In formulating problem Pi, note both the objective function as well as the constraint set can be conditioned upon the coordinating input, $\gamma_i(t+1)$. The manner in which conditioning occurs is dependent upon the decomposition approach that is used. The Dantzig-Wolfe algorithm (2) and Bender's partitioning algorithm (1) are two well-known methods.

In addition to selecting its optimal decision, the infimal must also define a set of responses, called the improvement set, $X''_i(t+1)$, such that

$$x_i \ll X''_i(t+1) \dashrightarrow f_i[x_i \mid \gamma_i(t+1)] \le f_i[x^*_i(t) \mid \gamma_i(t+1)] \tag{2}$$

while simultaneously ensuring any element of the improvement set $X''_i(t+1)$ must also satisfy any constraints currently associated with problem Pi. This improvement represents the primary feedback to the supremal. Given the previous reference model $X'_i(t)$ and the feedback $X''_i(t+1)$ for i=1,...,N, the supremal then updates its reference models using the mapping

$$\Pi: [X_i'(t), X_i''(t+1)] \dashrightarrow X_i'(t+1) \qquad (3)$$

Properties of Decomposition Algorithms

Feasibility. To ensure feasibility (5,9), the supremal's reference model for the infimal i's decision-space, $X_i'(t)$, must be contained within X_i for $i=1,\ldots,N$ and for all t. Thus, whenever $x_i^*(t)$ is selected from $X_i'(t)$, the feasibility with respect to the infimal's constraints is assured.

Convergence. Let $(x_1^0,\ldots,x_N^0)$ be the optimal solution to the original problem P0. For the decomposition procedure to achieve the optimum solution to P0, certain mathematical properties are required (5,9). To discuss these properties, it is beneficial to view the workings of the decomposition in progressing from one iteration to the next as the mapping given as

$$\delta: x_i^*(t) \longrightarrow x_i^*(t+1) \quad \text{for } i=1,\ldots,N. \qquad (4)$$

The mapping δ allows the successive generation of optimal solutions by the supremal on subsequent iterations to be viewed as a fixed point algorithm. This sequence converges provided δ satisfies

Property 1:
$x_i^*(t) = x_i^*(t+1)$ for all $i=1,\ldots,N$ if and only if
$x_i^*(t) = x_i^0$ for all $i=1,\ldots,N$,

Property 2:
If for some $i=1,\ldots,N$, $x_i^*(t) \neq x_i^*(t+1)$, then
$f[x_1^*(t),\ldots,x_N^*(t)] > f[x_1^*(t+1),\ldots,x_N^*(t+1)]$,

Property 1 says that the algorithm stops at the optimal solution only. Property 2 implies that the objective function will improve on each iteration. Since these convergence statements are identical to those of any fixed point algorithm no proof will be given.

Coordinability. Finally, decomposition algorithms must be "coordinable". This says that each hierarchical level must be able to both determine and effect its own optimum course of action with respect to the true problem P0. We use the definition found in (11).

If $x_i^*(t) \longrightarrow x_i^0$ for all $i=1,\ldots,N$
then $x_i^0(t) \longrightarrow x_i^0$ for all $i=1,\ldots,N$.

Succinctly stated, if the optimal solutions generated by the supremal are approaching the optimal solution to the problem P0, then the optimal solutions being generated by the individual infimals must also be approaching their corresponding components of the optimal solution to problem P0.

APPLYING DECOMPOSITION TO THE PS PROBLEM

A formal statement of the production scheduling problem was given in the preceding section. In developing a solution to this problem, we make two important and realistic assumptions. First, we assume that decision makers behave in a cooperative fashion to satisfy the overall goals. Second, we assume that individual process controllers (the infimals) possess more detailed information concerning the variables and constraints associated with their decisions than the interprocess coordinator (the supremal). These assumptions result in a downward flow of authority and an upward flow of aggregated feedback information.

These assumptions also imply that the interprocess coordinator (IPC) does not have the detailed information necessary to determine the exact sequence of activities at each process. Therefore, we propose to decompose the original production scheduling problem into two levels based on the approach developed in the preceding section. For this to work, we must expand that technique to include both multi-criteria decision-making and the stochastic nature of the manufacturing environment. Futhermore, we must impose consistent objectives at each level, and generate the information necessary to meet those objectives.

The Supremal

The supremal in the decomposition is the IPC. It will determine estimates for the earliest start time and latest finish time for JOB_j at P_n. These are designated E_{jn} and L_{jn} respectively. To do this, the IPC estimates the time required to complete JOB_j at P_n, t_{jn}, and optimizes the utility function

$$W[f^1(E_{11},\ldots,L_{jn}),\ldots,f^L(\cdot)]$$

subject to due dates and a set of coupling constraints.

The coupling constraints which interconnect the individual Process Controllers include the required precedence relationships that specify the order in which the processes must be applied to produce $PROD_m$ and any alternatives that may exist. Material transfer is another major concern since it determines the times between which a job will exit one process and arrive at the subsequent process. The flexibility introduced through automated transport devices and advanced processing technologies has made an exact mathematical formulation of all the coupling constraints extremely complex.

To estimate the times t_{jn}, the IPC builds a reference model, $g_{mn}'(\cdot)$, for the true probability density, $g_{mn}(\cdot)$, of time required to manufacture the $PROD_m$ corresponding to JOB_j on P_n. Sources for this model could be expected times and deviations from a process plan, empirical measurements from historical records, or a mathematical model based on process feedback. That model would first estimate the time required to complete each task in JOB_j on P_n, and sum them to get an estimate of t_{jn}.

It is important to note that the IPC does not have detailed timing information on the individual tasks, $TASK_{kjn}$, that makeup JOB_j at P_n. Therefore, it makes no attempt to stipulate start and finish times for those tasks. Rather, it is only concerned with the start and finish times of the entire job. Determining the exact sequence of tasks for each job and the time needed to perform those tasks are left to the individual Process Controllers (the infimals).

The Infimals

The infimals in the proposed decomposition are the process controllers, PC_n for $n=1,\ldots,N$, associated with the N manufacturing processes. The input to the subordinate PC_n is the set of two-tuples
$\{(E_{1n},L_{1n}),\ldots,(E_{jn},L_{Jn})\}$. Let us define

$$d_{jn} = L_{jn} - E_{jn} \qquad (5)$$

as the proposed maximum duration for the process P_n in acting upon JOB_j. The principle of decomposition implies that once this specification has been made, the PC_n retains full autonomy in the selection of the optimal processing sequence. Furthermore, this processing sequence provides an immediate action which the PC_n must then implement.

If t_{kjn} represents the actual time needed to complete $TASK_{kjn}$, then the infimal must generate a sequence of tasks, and start and finish times for those tasks (E_{kjn} and L_{kjn}) such that

$$\sum t_{kjn} < d_{jn} \qquad (6)$$

In general, t_{kjn} is a random variable with a probability density function, $g_{kmn}(t)$. Furthermore, it is important to note that $g_{kmn}(t)$ is never known with certainty as it is conditioned upon the state of the process, the performance quality of the preceding steps, and the properties of the input materials. It will be assumed, however, that the infimal or the PC_n will have the essential information to guide it in the selection of the processing steps which result in (6) above being satisfied. Under our assumptions of a cooperative organization it, will be assumed that the PC_n will take the action which is most consistent with the organization's goals.

THE ALGORITHM

We now present an approach which can be used to solve both the supremal and the infimal problems described above. As indicated, the complex nature of the constraints, the stochastic nature of the manufacturing environment, and the presence of multiple but conflicting objectives, make a precise mathematical programming formulation of and solution to these problems very difficult. An alternative, and often used approach involves the specification of a realistic simulation. Off-line simulation studies (12,13) have gained considerable acceptance, as a means generating good solution to the PS problem.

This study proposes to use simulation to provide a real-time, on-line scheduling tool for both the supremal and infimal problems. The schematic for the proposed approach is given in Figure 2. In the discussion that follows, examples are taken from the supremal problem. An obvious substitution of subscripts, will yield similar results for the infimals.

Planning

The planning elements include the selection of evaluation criteria and scheduling rules, simulations, statistical analysis, and compromise analysis.

Evaluation Criteria. The evaluation criteria can be a combination of goals related to the performance of the entire manufacturing system, some or all of the processes, and some or all of the jobs. These criteria are often fixed, and set by management. However, they can also be a function of the current state of the system and changed each time a new schedule is required.

Scheduling Rules. The scheduling rules can be a combination of preselected job release strategies, queuing strategies, material handling strategies, and any number of well-known dispatching rules. As with the evaluation criteria, these rules can be fixed or vary with the state of all or part of the system.

Simulations. A separate, independent processor is dedicated to running a simulation for each potential scheduling rule. The simulation is integrated with shop floor data collection systems so that each trial can be initialize to the current state of the manufacturing system. Assuming that a total of K simulation trials is performed for each scheduling rule, the following table could be generated:

k	Simulation Results
1	$E_{11}^{1}\ L_{11}^{1} \cdots E_{1N}^{1}\ L_{1N}^{1} \cdots E_{J1}^{1}\ L_{J1}^{1} \cdots E_{JN}^{1}\ L_{JN}^{1}$
$\vdots$	$\vdots\ \vdots\ \ \vdots\ \vdots\ \ \vdots\ \vdots\ \ \vdots\ \vdots$
K	$E_{11}^{K}\ L_{11}^{K} \cdots E_{1N}^{K}\ L_{1N}^{K} \cdots E_{J1}^{K}\ L_{J1}^{K} \cdots E_{JN}^{K}\ L_{JN}^{K}$

Table 1 — Simulation Results for each Scheduling Rule

Given this table for each scheduling rule, each of the L objectives is evaluated giving

$$f_k^l = f^l(E_{11}^k \ldots L_{JN}^k) \text{ for } k=1,\ldots K \text{ and } l=1,\ldots L. \qquad (7)$$

As an example, the tardiness of a given JOB_j could be computed as

$$T_j^k = \max_n(0, \max[L_{jn}^k] - D_j) \text{ for } j=1,\ldots,J \text{ and } k=1,\ldots K. \qquad (8)$$

Statistical Analysis. Having evaluated the objective functions for each of the simulated cases, a statistical analysis (10) is performed on the data. Specifically, for each objective $f^l(\cdot)$ for $l=1,\ldots,L$ and scheduling rule r R an empirical probability density function will be developed giving

$$Pr[f_r^l(\cdot) \leq z] = F_r^l(z) \qquad (9)$$

The following statistics can then be computed as

$$\bar{f}_r^l = \text{Mean or Ex } [f_r^l] \qquad (10)$$

$$(s_r^l)^2 = \text{Sample Variance or Ex } [f_r^l - \bar{f}_r^l] \qquad (11)$$

$$m_r^l = \text{Minimum or } \min_k \,[f_k^l|r] \qquad (12)$$

$$M_r^l = \text{Maximum or } \max_k \,[f_k^l|r] \qquad (13)$$

Compromise Analysis. Having developed the essential statistics associated with each analyzed objective, conditioned upon the selected scheduling rule, the next step is to develop the best compromise scheduling rule, r^*. First we determine the nondominated (4) set of scheduling rules, denoted by R^*. The set R^* will be defined here such that $r \in R^*$ if for every $r' \in R$ there exists an $l \in [1,\ldots,L]$ such that

$$\bar{f}_r^l \geq \bar{f}_r^l{}' \qquad (14)$$

Given the nondominated rule set R^*, the next step is to determine the minimum for each objective function over R^* as

$$n^l = \min_{r \in R^*} \,(m_r^l) \qquad (15)$$

Please note that the minimization in equation (15) is over the nondominated set of rules, R^*, only. In a similar fashion, the maximum for each objective over R^* is next determined as

$$M^l = \max_{r \in R^*} \,(M_r^l) \qquad (16)$$

In this manner, the range of compromise for each objective f^l

over R^* is defined as the interval $[m^l, M^l]$. Using the statistics for the nondominated strategies R^* and the associated range of compromise, the "best" compromise strategy $r^* \in R^*$ is then chosen.

Control

Having found r^*, we now focus on the control functions: list generation and coordination.

List Generation. The first major control function generates an event list which the supremal will attempt to implement. Using the current state of the processes P_n $(n=1,\ldots,N)$ with the selected best compromise rule r^*, an additional single pass of the simulation is made. For this simulation, meaningful or realistic values for each random duration d_{jn} must be selected. That is, the values of d_{jn} will be chosen such that using $g'_{mn}(\cdot)$ there is a minimum specified probability for completing the task outside the selected duration. This simulation will result in the event list

$$\underline{E} = [E_{11}\ L_{11}, \ldots, E_{JN}\ L_{JN}] \tag{17}$$

The event list $\underline{E}$ is then sorted into three other lists. $\underline{E}$ is sorted chronologically into a master schedule $\underline{T}$, by JOB into a scheduling list $\underline{J}$, and by process into a list $\underline{C}$. The job scheduling list $\underline{J}$ provides the information necessary to track each JOB_j at any given time. The process scheduling control list $\underline{C}$ will permit the prediction of the status of a given process P_n at any given time.

Coordination. The event lists $\underline{T}$, $\underline{J}$ and $\underline{C}$ provide the data needed by the supremal to 1) coordinate the activities of the subordinate infimals, and 2) provide feedback status to the next higher level in the hierarchy on job completion. The coordination comes from the coupling constraints arising from the precedence relationships and the material handling considerations. With respect to these constraints, it is thus assumed that $\underline{T}$ represents a feasible solution. The primary uncertainties are the process durations d_{jn}. Under the assumption of a cooperative hierarchy, it will now be assumed that each PC_n $(n=1,\ldots,N)$ will actively attempt to fit the process duration d_{jn} within the time interval $[E_{jn}, L_{jn}]$.

Outputs to Infimals

There are two outputs that the supremal presents to each infimal for planning purposes: a list of jobs and their associated start and finish times, and a set of objectives. The start and finish times for each job is contained in the two-tuple (E_{jn}, L_{jn}). In addition, each infimal must be presented with the current $f_1^l(E_{1n}, L_{1n}, \ldots, E_{Jn}, L_{Jn})$, representing the infimal's contribution to the supremal's objective function $f^l(\cdot)$ for $l=1,\ldots,L$. The infimal n's objective functions $f_i^l(\cdot)$ must be consistent with the supremal's objectives. The procedures used in choosing these functions are under development.

Conflict Resolution

The feedback information from the PC_n, (E'_{jn}, L'_{jn}) gives E'_{jn} as the actual initiation time and L'_{jn} as the predicted completion time for JOB_j. The actual processing time is given by

$$t'_{jn} = L'_{jn} - E'_{jn} \tag{13}$$

In general, t_{jn} will not equal d'_{jn}. Whenever this happens, the event list $\underset{\sim}{T}$ is no longer valid since JOB_j will not be completed at the scheduled time. In this case, $\underset{\sim}{T}$ must be updated to reflect this discrepancy. This requires the supremal to update its solution, in real-time, to restore feasibility. The updating of the optimal response from the PC_n into the current solution for the supremal provides an interaction within decomposition that has not been extensively explored. A two step process is envisioned.

First, we must determine the impact of the discrepancy on the current schedule. The output of this analysis will determine whether the current compromise policy, r^*, and the resulting schedule is still realizable. If it is, then we simply update the estimates for the expected durations, d_{jn}, and generate a new lists $\underset{\sim}{T}$, $\underset{\sim}{J}$, and $\underset{\sim}{C}$. If r^* is no longer valid, then we must go through the entire exercise again. We are in the process of defining quantitative measures for deciding when the current schedule cannot be met.

SUMMARY

This paper has presented an algorithm to address the two phases of the real-time production scheduling problem: planning and control. It combines both discrete event simulation and decomposition of mathematical programming in ways which advance the state-of-the-art of both fields. In addition, it allows for the potential of integrating symbolic computing techniques from Artificial Intelligence with quantitative methods from Operations Research. Work has begun on implementing the various components of this algorithm.

REFERENCES

1. Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numersche Mathematik, Vol. 4, No. 3, 1960, pp. 238-252.

2. Dantzig, G. B. and Wolfe, P., "Decomposition Principles for Linear Programs," Management Science, Vol. 8, No. 1, 1960, pp. 101-111.

3. Davis, W., "Decision-Making and Control Hierarchies for Production Systems", Report No. 145 - PLAIC, Purdue University, West Lafayette, Indiana, 1984.

4. Dessouky, M. I., Ghiassi, M. and Davis, W. J. "Estimates of the Minimum Nondominated Criterion Values in Multiple-Criteria Decision-making," Engineering Costs and Production Economics, Vol. 10, 1986, pp. 95-104.

5. Geoffrion, A., "Elements of Large-Scale Mathematical Programming, Parts I and II", Management Science, Vol. 16, No. 11, pp. 652 - 691, 1970.

6. Graves, S, C., "A Review of Production Scheduling," Operations Research, Vol. 29, pp, 646-675.

7. Jackson, R., and Jones, A., "An Architecture For Decision-Making in The Factory of the Future", (to appear), ORSA Interfaces Special Issue on Flexible Manufacturing.

8. Jones, A. and Whitt, N. (eds.), "Proceedings of Factory Standards Model Conference", National Bureau of Standards, November, 1985.

9. Lasdon, L., Optimization Theory for Large Systems, Macmillan, New York, 1970.

10. Law, A. M. and Kelton, W. D., Simulation, Modeling and Analysis, McGraw-Hill, New York, 1982.

11. Mesarovic, M. D., Macko, D. and Takahara, Y., Theory of Hierarchical, Multilevel Systems, Academic Press, New York, 1970.

12. Miles, T., Erickson, C. and Batra, A., "Scheduling a Manufacturing Cell with Simulation," Proc. of 1986 Winter Simulation Conference, Washington, D.C., pp. 668-676.

13. Norman, T. A. and Norman, V. B., "Interactive Factory Scheduling Using Discrete Event Simulation, Proc. of 1986 Winter Simulation Conference, Washington, D.C., pp. 665-667.

14. Raman N., "A Survey of the Literature on Production Scheduling as it Pertains to Flexible Manufacturing Systems", National Bureau of Standards Report NBS-GCR-85-499, Gaithersburg, Maryland, 1985.

15. Stecke, K. E., "Design, Planning, Scheduling and Control Problems of Flexible Manufacturing Systems," Proc. of Flexible Manufacturing Systems Conference, First ORSA/TIMS Special Interest Conference, Ann Arbor, Michigan, Aug. 1984.

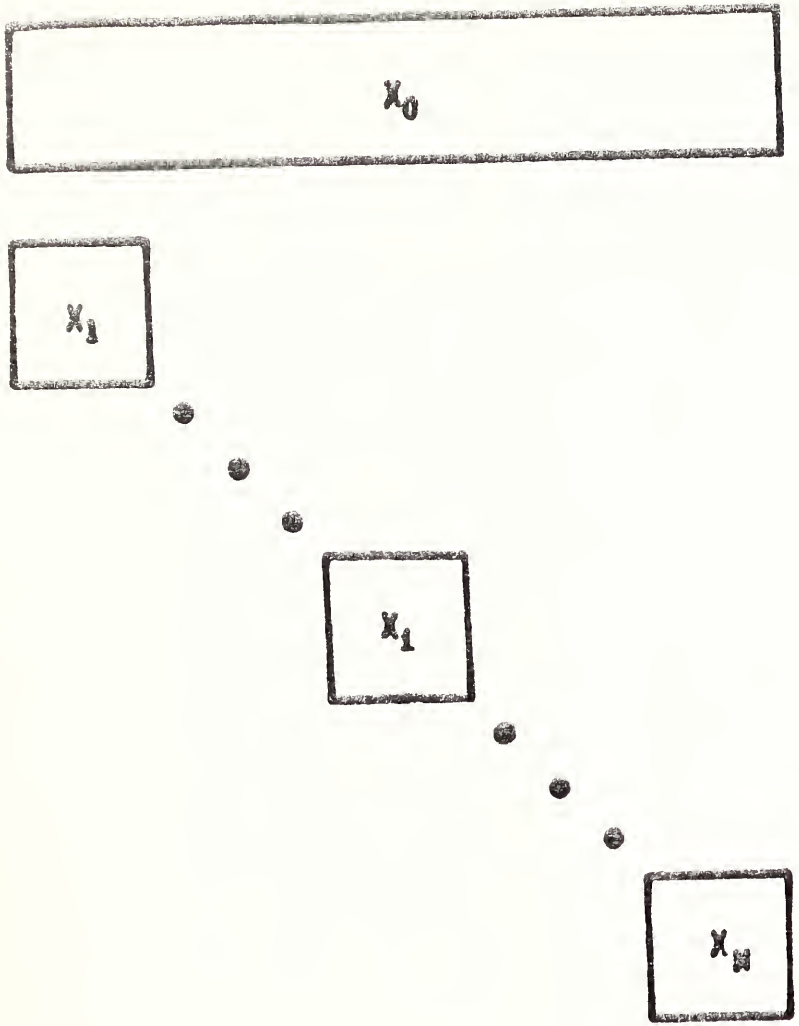

FIGURE 1a. BLOCK ANGULAR STRUCTURE

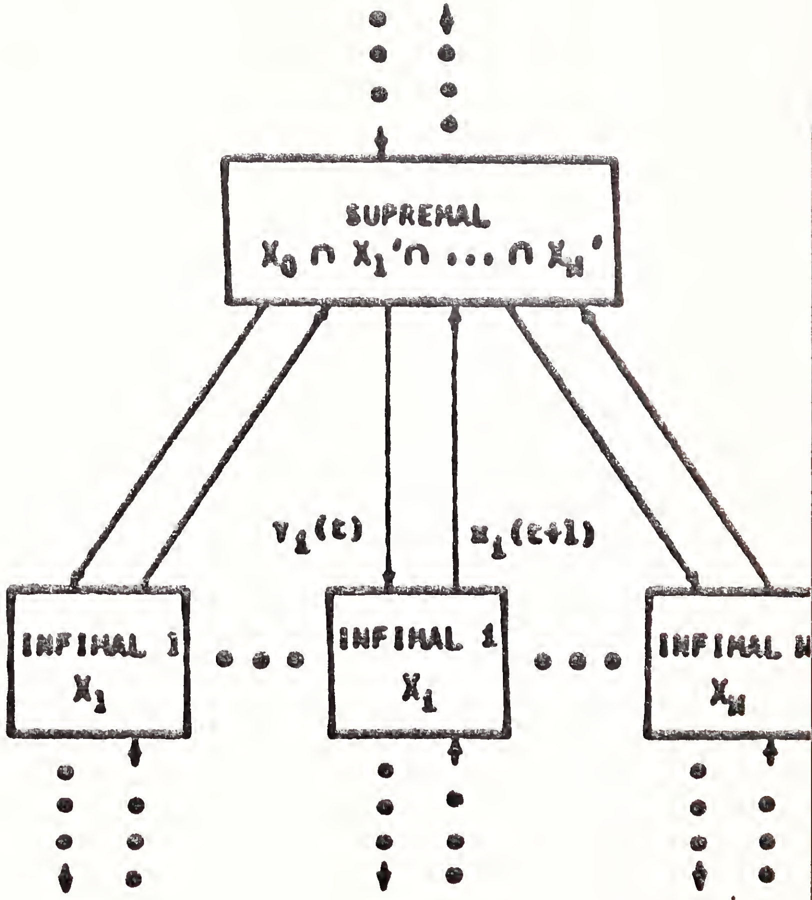

FIGURE 1b. HIERARCHICAL STRUCTURE

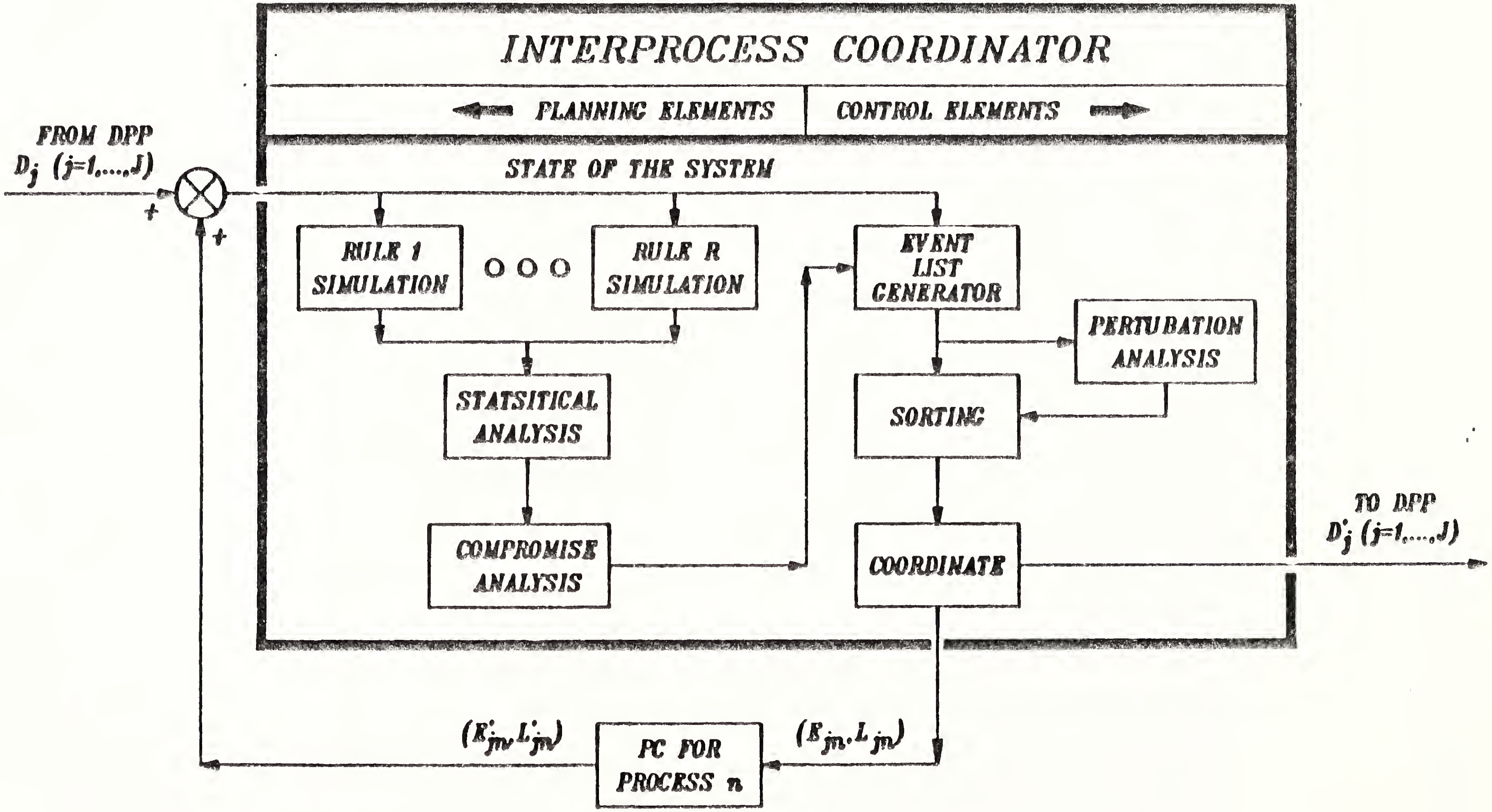

FIGURE 2. STRUCTURE FOR PRODUCTION SCHEDULER